UCRL-TR-237205

Application of the PSUADE tool for Sensitivity Analysis of an Engineering Simulation

H. Hsieh

December 11, 2007

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Application of the PSUADE Tool for Sensitivity Analysis of an Engineering Simulation

Henry (Shang-Rou) Hsieh

Structural Mechanics Group
New Technologies Engineering Division
Lawrence Livermore National Laboratory

CODTU-2006-0029

ASC™

Table of Content

List of Figures

List of Tables

Acknowledgement

I would like express my gratitude to the following persons who had provided support in preparing this report: Mark Garcia, Frank Magness, Anthony Depiero, Steven Chidester, Tony Lee and Robert Ferencz. Many extensive and interesting discussions with Charles Tong, the developer of PSUADE, made this report possible and fun to write.

1. Introduction

Due to the advent of computer technology, much of today's engineering analyses rely on computational tools. Despite steady advances in computing power, the computational expense in conducting many calculations remains unaffordable or unfeasible. One way to explore a model's functional relationship between its set of input parameters, ***X,*** and the vector of output response, ***Y,*** is to apply statistical techniques with computer simulation. PSUADE is being developed to facilitate such explorations.

This report focuses on using PSUADE[1] as a tool for (global/local) sensitivity analysis and uncertainty quantification. The Steven Impact Test which studies high explosive initiation subjected to low speed impact is chosen as an illustrative example. However, the investigation of HE initiation mechanisms is beyond the scope of this report. Please refer to [1-6] for a more detailed and technical content study for Steven Impact Test. The remainder of this report is organized as follows: Section 2 contains a brief review of the Steven Impact Test. Section 3 reviews a general flowchart in conducting uncertainty quantification. Section 4 gives an overview of PSUADE. Application examples of PSUADE are described in section 5. Section 6 contains concluding remarks regarding PSUADE.

[1] **P**roblem **S**olving Environment for **U**ncertainty **A**nalysis and **D**esign **E**xploration

2. Steven Impact Test

The main purpose of a Steven Impact Test (depicted in Figure 1) is to study the initiation of high explosive (HE) subjected to impact by projectile with low velocity. There are several HE initiation mechanism models, such as reactive flow or frictional work, that have been proposed in the past. It is not the intent of this report to study the validity of HE initiation mechanisms or propose new ones. In this report, frictional work is simply chosen to be the governing initiation mechanism to illustrate the application of PSUADE. We are going to study how the frictional work get affected by the following parameters: projectile velocity, stiffness of the cover plate, friction coefficients between HE and its surrounding materials, tangential hardening modulus, yielding stress, and shear modulus of the HE.

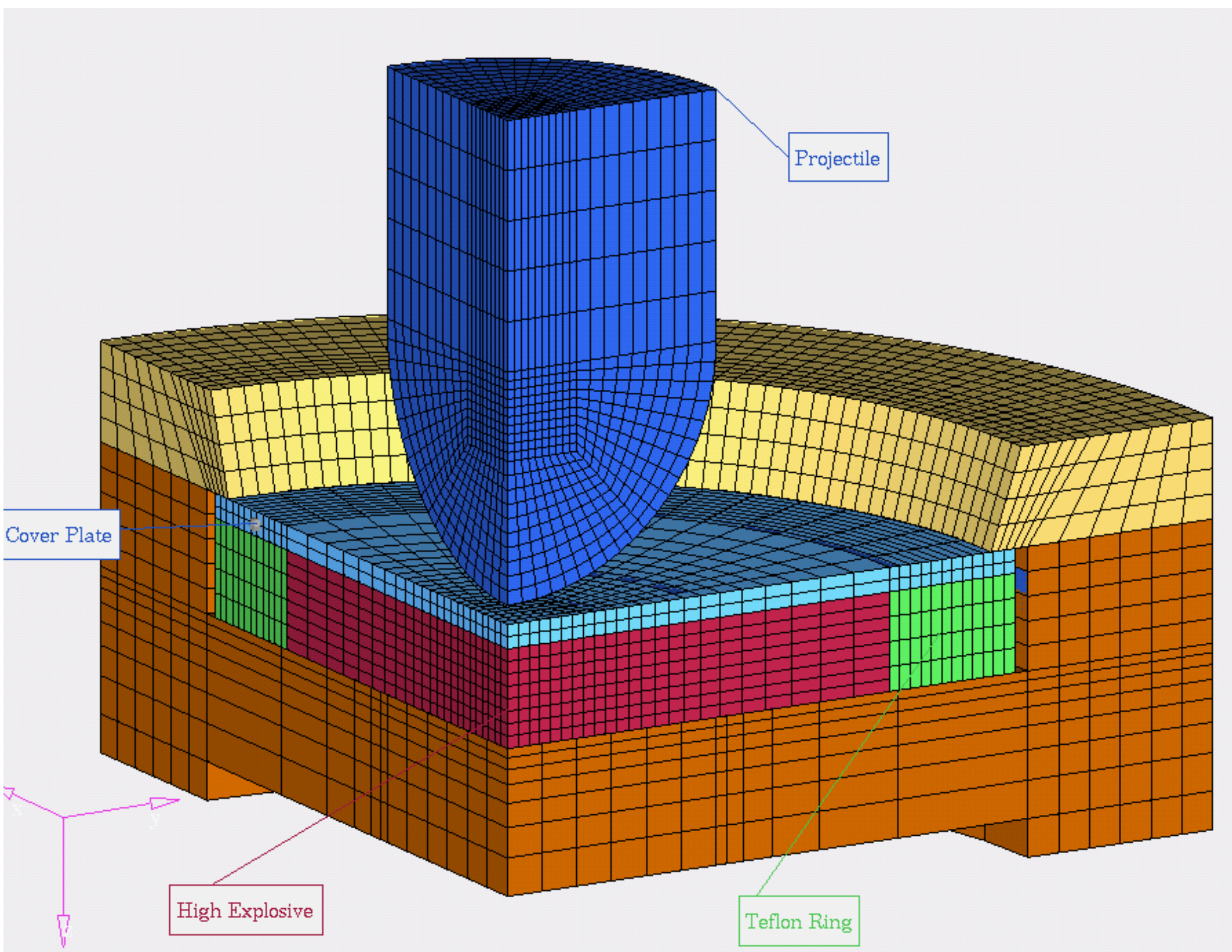

Figure 1: Schematic Diagram of Steven Impact Test

3. Overview of Analysis Procedure

Figure 2 shows a generic analysis procedure leading to uncertainty quantification (UQ) using a response surface. (Other techniques, such a direct Monte Carlo, are perhaps conceptually more straightforward but not considered computationally tractable for the models anticipated in W Program applications.) The procedure consists of the following steps: Simulation Model, Parameter-Diagram (P-Diagram), Screening Experiment, Response Surface, Model Validation, and Uncertainty Quantification.

A complete understanding of the relevant system requirements is necessary to successfully develop an analytical model. The requirements must be sufficiently defined to establish the scope of the problem, to understand the goal of the analysis, and to identify necessary physics to be included.

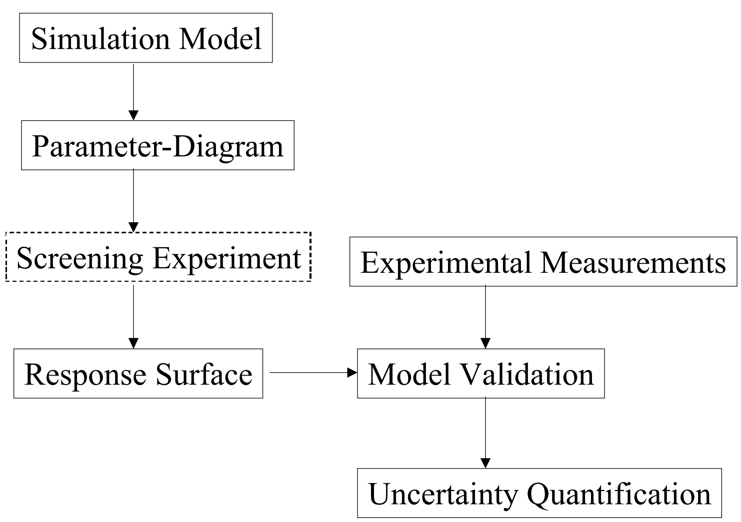

Figure 2: Overview Analysis Flowchart for Uncertainty Quantification

<u>Parameter Diagram (P-Diagram)</u>

P-Diagram is a must for every development project. It is a way of succinctly defining the development scope. This step in the UQ analysis flowchart (Figure 2) is to identify each factor being included in the parameter-diagram. This can only be done efficiently by reviewing the content of the analytical model, the intent of analysis, and the nature of the application. During the course of any computational modeling certain assumptions are made. Therefore, it is very important to review the content and the capability of the resultant analytical model (the simulation model) in defining the P-diagram.

Figure 3 shows a P-Diagram for the Steven Impact Test. The term "control factor" is used to designate factors which can be controlled and affect the system response. Typical examples of control factors are: material selection, thickness, and design feature. For the Steven Impact Test, the following parameters are chosen to be the control factors: yielding stress of explosive (σ_y), shear modulus of explosive (G), tangential hardening modulus of high explosive (E_h), yielding stress of cover plat (K), and multiple coefficients of friction between explosive and surrounding materials (μ_3, μ_6, and μ_7). The term "input factor" is used to describe the measurable causal agents to which the system will respond. For the Steven Impact Test, this input factor is the projectile velocity. The "output factor" is the response of the system. For Steven Impact Test, this output response (factor) is frictional work (frw). In the original definition [8], the factors which cannot be controlled are categorized as "noise factors", including (1) piece-to-piece interaction/ variation, (2) wear/fatigue, (3) product duty cycle, (4) environment (such as climate), and (5) systems interaction. In computer experiment models, the definition of such noise factor needs to be refined. Contrast to the conventional experimental environment, there is no randomness in computational environment. All the parameters, boundary conditions, … etc, are deterministic. The noise factor in computational environment is probably best described as "known-unknown" and "unknown-unknown".

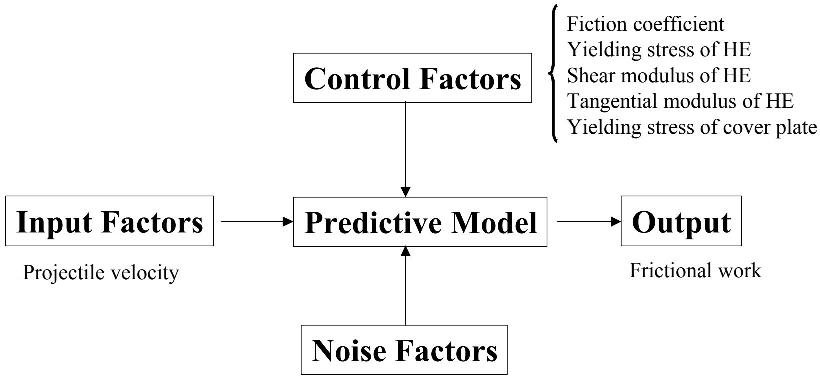

Figure 3: Parameter Diagram (for Steven Impact Test)

After the P-diagram has been established, one needs to define the design space for the study. This is equivalent to define the parameter ranges of input and control factors (Figure 3). For the Steven Impact Test studied here, the parameter range of each of the control factor is listed in the following table:

Table 1: List of Parameters and Ranges for Steven Impact Test

Parameter	Description	Range [min, max]
K	Yielding stress of cover plate	[6.2964E-3, 7.6956E-3]
G	Shear modulus of HE	[3.519E-2, 4.301E-2]
σ_y	Yielding stress of HE	[7.4466E-4, 9.1014E-4]
E_h	Tangential Hardening Modulus	[5.796E-5, 7.094E-5]
μ_3	Friction coefficient between HE and cover plate[7]	[0.20, 0.23]
μ_6	Friction coefficient between HE and Telfon [7]	[0.23, 0.27]
μ_7	Friction coefficient between HE and backplate[7]	[0.20, 0.23]

Screening Experiments

This step is optional. The screening experiment is critical when either the numbers of rote combinatory analyses are large or each analysis is very expensive. The central question of the screening experiment in the context of modeling and computer simulation is: Which factors – among the many potentially important factors – are really important? One of the aims in screening is to come up with a short list of important factors, and typically a risk-based graded approach is utilized to establish a parameter hierarchy in terms of the importance to the system outcome. Screening methods are created to deal with models containing hundreds of input factors. For this reason, these methods must be economical. There exists a trade-off between computational cost and information obtained from these methods.

Response Surface

The response surface is also known as a meta-model or surrogate model. The first step in meta-model building is to select an appropriate design of (computational) experiment. After all the necessary computer simulations having been completed, the next step is to choose an approximation model and then to fit the model. However, the functional form of the meta-model dictates the experimental design. One shall always keep this in mind when building the meta-model. Typically, one chooses low-order polynomials to approximate the response of the model and fit the model with a linear least-square regression scheme. If *physical* experimental data are available, one shall validate the predictive model before proceeding to uncertainty quantification. Model validation, a rich subject, is beyond the scope of this report.

Uncertainty Quantification

The uncertainty quantification is conducted based on the surrogate model that was created in the previous step. This is intended to be equivalent to studying the propagation of uncertainty through the original (simulation) model. To study the probabilistic structure of a model's response, one simply applies a space-filling sampling scheme, such as Latin-Hypercube or Monte Carlo, to all the meta-model parameters (input and control factors) with corresponding probabilistic structures.

4. Overview of PSUADE

This section provides a brief overview of PSUADE[15]. PSUADE is an analysis tool created to facilitate model validation, including global/local sensitivity analysis and uncertainty quantification.

Structure of PSUADE

Figure 4Figure 4 depicts PSUADE's execution model. In this model, PSUADE includes the following elements: control, sample generator, and analysis. PSUADE oversees (i.e. controls) the entire calculation process for uncertainty quantification. Through the sample generator, PSUADE will create a design matrix to study the response of the system with simulation model. In the analysis phase, PSUADE will the collect the simulation results, assess sensitivity analysis and conduct uncertainty quantification.

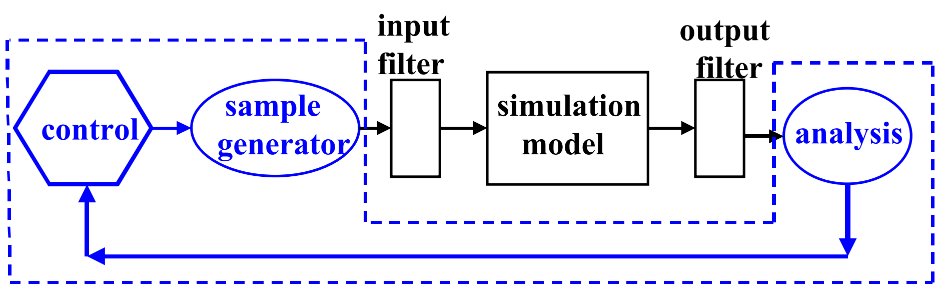

Figure 4: PSUADE's Execution Model

The term input filter can be described as "pre-processing"; whereas output filter is "post-processing". Input and output filters, together with the simulation model, depend upon the application example. Users shall provide their own scripts (e.g. Perl, Python, …..) and programs to accomplish the needs for pre- and post-processing. Appendix A lists an example written in Python covering input and output filters.

PSUADE interacts with users via a few files. At the first level, a PSUADE input file (called `psuade.in` here) has to be provided via the command line. Inside this input file the users have to specify two files where the sample data and the results are to be communicated.

Structure of `psuade.in` File

The content of the initial setup file of PUSADE consists of the following sections: INPUT, OUTPUT, METHOD, APPLICATION, and ANALYSIS. Table 2 lists the functionality of each section:

Table 2 PUSADE section name and functionality

Section Name	Functionality
INPUT	Defines number of inputs, their names, ranges, and distributions.
OUTPUT	Defines numbers of output and their names
METHOD	Specifies the sampling method and corresponding settings
APPLICATION	Set up runtime parameters
ANALYSIS	Specifies type of analysis and corresponding settings

Depending upon particular applications, the user of PSUADE can modify the content of each section to reflect the needs of his (or her) analysis. Appendix A lists an example of this interface file. In addition, users have to specify in the input file the driver for the computational model to be evaluated (details given in the APPLICATION section described below).

Structure of `psuadeData` File:

Comparing with the initial setup file, **psuadeData** consists of one additional section: PSUADE_IO. This section contains all the sample points and resulting data points. The first line in this section consists of three numbers: the first field is to define the number of input parameters, the second field is for the number of output parameters, and the last field stands for the number of simulation runs. The rest of the PSUADE_IO section contains the design matrix, which will be used to sample the simulation, and simulation results associated with sampling points. Appendix B lists an example of the `psuadeData` file.

Capability of PSUADE

The current version (1.0) of PSUADE is capable of:

- Uncertainty quantification
- Global sensitivity analysis (qualitative and quantitative)
- Higher order sensitivity analysis (interactions)
- Confidence interval analysis / hypothesis testing / Validation
- Design exploration
- Response surface analysis (model fitting)
- Model reduction (set research priorities)
- Design optimization/calibration/parameter estimation

The term 'design exploration' stands for the application of different sampling schemes, including Latin-Hypercube[11], Morris One-At-a-Time[11], (Fractional) Factorial Design [8, 9]. Global and high sensitivity analyses are useful for screening experiment. Unlike the traditional definition [8,9], the response surface in PSUADE is referred to a much general model fitting (also known as metamodel)[16]. Typically, confidence interval analysis, hypothesis test, and model validation are conducted using the metamodel.

5. Application Examples

This section describes the application of PSUADE. The applications of PSUADE are presented using several case studies. These examples are presented to show a step-by-step application of PSUADE for uncertainty quantification. Due to lack of conclusive experimental data, model validation is not included in this report. This section starts with an example to show the initial setup of PSUADE. The second example illustrates how to apply PSUADE to create an experimental design. Sensitivity analyses, an important subject in uncertain assessment, are demonstrated in the third example. The last example is for uncertainty quantification. Notice, these examples are created to reflect the step-by-step operation involved in uncertainty quantification.

Example 1: Initial setup for PSUADE

This section lists the steps involved in using PSUADE to generate a design matrix used later. The following are the screen dumps from exercising PSUADE. To distinguish from regular documents, all the screen outputs and user's inputs in using PSUADE are expressed with *italic* style. All the user inputs are highlighted by ***bold-italic*** to distinguish from PSUADE output message. If one follows the procedure as listed below, PSUADE will generate an input file, named `example1` (listed in Appendix B: Content of example1) in this case. After keying in ***psuade*** and pressing <enter> the following message shall appear on your monitor:

```
*** ************************************************** ***
*** Welcome to PSUADE main program (version 1.0)       ***
*** ************************************************** ***
PSUADE - A Problem Solving environment for
         Uncertainty Analysis and Design Exploration
```

The next step is to create an input file for PSUADE:

```
psuade> genfile example1
```

The following few steps are to create the design matrix for seven input parameters and one output parameters:

```
Enter the number of inputs (> 0) : 7
name for input 1 ? k
```

```
lower bound for input 1 ? 6.2964E-3
upper bound for input 1 (> lower bound)? 7.6956E-3
name for input 2 ? g
lower bound for input 2 ? 35.19E-3
upper bound for input 2 (> lower bound)? 43.01E-3
name for input 3 ? sigma_y
lower bound for input 3 ? 7.4466e-4
upper bound for input 3 (> lower bound)? 9.1014e-4
name for input 4 ? eh
lower bound for input 4 ? 5.796e-5
upper bound for input 4 (> lower bound)? 7.094e-5
name for input 5 ? mu_3
lower bound for input 5 ? 0.20
upper bound for input 5 (> lower bound)? 0.23
name for input 6 ? mu_6
lower bound for input 6 ? 0.23
upper bound for input 6 (> lower bound)? 0.27
name for input 7 ? mu_7
lower bound for input 7 ? 0.20
upper bound for input 7 (> lower bound)? 0.23
Enter the number of outputs (> 0) : 2
name for output 1 ? frw31
name for output 2 ? frw34
```

PSUADE will list all the available sampling schemes, including MC (Monte Carlo), FACTORIAL, LH (Latin Hypercube), OA (orthogonal Array), OALH (Orthogonal Array Latin Hypercube), MOAT, … etc.

```
available methods :
 MC    - Monte Carlo (random)
 FACT  - full factorial
 LH    - Latin Hypercube
 OA    - Orthogonal Array
 OALH  - OA-based Latin Hypercube
 MOAT  - Morris one at a time
 LPTAU - A Pseudo-random sequence
 METIS - A space-filling design
 FAST  - Fourier Amplitude Sampling Test
 FF4   - Fractional Factorial with Resolution IV
 FF5   - Fractional Factorial with Resolution V
sampling Method ? LH
number of replications (>= 1) ? 1
sample size (multiple of 1) ? 20
Do you want to add noise to the sample? (y or n) n
Please enter the absolute path for the following.
Where is your script? (enter NONE if not needed) NONE
Would you like to create the sample file? (y or n) y
```

```
The sample matrix is now stored in psuadeData file.
You can also use genmars command to convert the sample
matrix to a row-column format.
psuade> quit
```

After this, PSUADE will create a file named **example1** (listed in Appendix B) which serves the initial setup (or interface) file. Since we had chosen to create a sample file, there shall be additional file **psuadeData** created as well. If one decides not to use PSUADE as the engine for subsequent operation and analyses, including job submission and data collection, the execution of PSUADE ends. Appendix C lists the content of ***psuadeData*** based on **example1.** If one chooses not to generate a sample file, then there will be no psuadeData file.

Example 2: Sensitivity Analysis

Assuming all the analysis simulations having been completed and the response data been collected, the next step is to study the simulation model, e.g., its parametric sensitivity, response surface, and uncertainty quantification. ***Before proceeing to study the model, one shall rename the psuadeData since PSUADE will generate a new psuadeData and save the original datafile as psuadeData.sav.*, with * being the version of the saved file.*** For the sensitivity analysis, different methods are applied. The results from analyses are summarized in Table 3. Unless stated otherwise, all the following examples are based on data file, `psuade010705.dat` (listed in Appendix D).

```
*** ************************************************** ***
*** Welcome to PSUADE main program (version 1.0)        ***
*** ************************************************** ***
PSUADE - A Problem Solving environment for
         Uncertainty Analysis and Design Exploration
psuade> load psuade010705.dat
load complete : nSamples = 20
```

To look into the analysis tools available for uncertainty quantification, let us type in `**help uqsa**` and then press <enter>. PSUADE will list all available commands for uncertainty quantification on the screen:

```
psuade> help usaq
Commands for UQ/SA:
         splot   (scatter plot in matlab)
         uq      (UQ + matlab plot)
         ca      (correlation analysis)
         me      (main effect study + matlab plot)
         ie      (interaction effect study)
         int     (numerical integration)
         meplot  (plot main effects with Pgplot)
         meplot2 (plot main effect/interaction with Pgplot)
         rsplot  (plot response surface with Pgplot)
         rs      (2-input response surface in matlab)
         rs3     (3-input response surface in matlab)
         rs3m    (3-input response surface in matlab (movie)
         rsi     (2-input RS intersections in matlab)
         rsi3    (3-input RS intersections in matlab)
         smplot  (Morris' stdev-mean plot in matlab)
         rscheck (check quality of RS)
psuade> printlevel 2
psuade printlevel set to 2
psuade> rscheck
Enter output number (1 - 2) = 1
Which response surface tool would you like ?
0. MARS
1. Linear regression
2. Quadratic regression
3. Cubic regression
4. Quartic regression
5. Artificial Neural Network
6. User regression
Please enter your choice ? 0
You may want to transform the output as follows:
0. no transformation.
1. log transformation on all the inputs.
2. log transformation on all the outputs.
3. log transformation on all inputs and outputs.
4. log transformation on selected inputs.
5. log transformation on selected inputs and outputs.
Please enter your choice ? 0
RSFA:: MARS model.
RSFA:: output maximum norm          = 1.450000e-05
* ============================================== *
* MARS importance measure   1 = 0.000e+00
* MARS importance measure   2 = 2.612e+01
* MARS importance measure   3 = 0.000e+00
* MARS importance measure   4 = 0.000e+00
* MARS importance measure   5 = 0.000e+00
```

```
* MARS importance measure   6 = 0.000e+00
* MARS importance measure   7 = 1.000e+02
* ============================================== *
PSUADE : refine check = 1.457e-08 <? 1.000e+00
```

Using MARS, a non-parametric regression technique, the qualitative analysis result (importance measure) indicates that output response #1 (frw31) is very sensitive to the second (the shear modulus of HE) and the seventh (coefficient of friction between HE and backplate) input parameters. Next, we study the parametric sensitivity using the parametric regression technique, linear regression.

```
psuade> rscheck
Enter output number (1 - 2) = 1
Which response surface tool would you like ?
0. MARS
1. Linear regression
2. Quadratic regression
3. Cubic regression
4. Quartic regression
5. Artificial Neural Network
6. User regression
Please enter your choice ? 1
You may want to transform the output as follows:
0. no transformation.
1. log transformation on all the inputs.
2. log transformation on all the outputs.
3. log transformation on all inputs and outputs.
4. log transformation on selected inputs.
5. log transformation on selected inputs and outputs.
Please enter your choice ? 0
RSFA:: linear regression model.
RSFA:: output maximum norm          = 1.450000e-05
************************************************************
*************** Linear Regression Analysis *****************
* R-square gives a measure of the goodness of the model.   *
* R-square should be close to 1 if it is a good model.     *
*----------------------------------------------------------*
* Regression:: LS average error =   3.1483e-08 (max=  1.4500e-
05)
*               coefficient   std. error   t-value
* Constant   =   8.9136e-06   5.1418e-07   1.7335e+01
* Input   1 =  -9.8530e-05   2.5080e-05  -3.9286e+00
* Input   2 =  -2.6403e-05   4.5539e-06  -5.7980e+00
* Input   3 =   1.3076e-03   2.2762e-04   5.7447e+00
```

```
* Input   4 =  -2.4624e-03   2.6060e-03  -9.4489e-01
* Input   5 =   6.0435e-06   1.2245e-06   4.9356e+00
* Input   6 =   2.5741e-06   8.4092e-07   3.0610e+00
* Input   7 =   1.8603e-05   1.2174e-06   1.5281e+01
* Regression model R-square =   9.6870e-01
*==========================================================*
***** Standardized Regression Coefficients (SRC) ***********
* When R-square is acceptable (order assumption holds), the*
* absolute values of SRCs provide variable importance.     *
*----------------------------------------------------------*
* based on nSamples = 20
* Input   1 =  -2.2337e-01
* Input   2 =  -3.3453e-01
* Input   3 =   3.5059e-01
* Input   4 =  -5.1386e-02
* Input   5 =   2.9376e-01
* Input   6 =   1.6682e-01
* Input   7 =   9.0421e-01
*==========================================================*
PSUADE : refine check = 7.040e-09 <? 1.000e+00
```

Next, let us try to fit a regression model with quadratic polynomial model,

```
psuade> rscheck
Enter output number (1 - 2) = 1
Which response surface tool would you like?
0. MARS
1. Linear regression
2. Quadratic regression
3. Cubic regression
4. Quartic regression
5. Artificial Neural Network
6. User regression
Please enter your choice ? 2
You may want to transform the output as follows:
0. no transformation.
1. log transformation on all the inputs.
2. log transformation on all the outputs.
3. log transformation on all inputs and outputs.
4. log transformation on selected inputs.
5. log transformation on selected inputs and outputs.
Please enter your choice ? 0
RSFA:: quadratic regression model.
RSFA:: output maximum norm          = 1.450000e-05
* Regression: reduce order to 1 - not enough samples (36,20).

(the rest of output message from PSUADE are skipped)
```

The order of polynomial gets reduced to 1 (i.e. linear polynomial model), since it requires 36 sampling points for a complete quadratic polynomial model. However, users can define their own polynomial model and then apply PSUADE for regression study. To do so, the user needs to create the file `user_regression_file` (listed in Appendix E), which defines the polynomial model for regression analysis (so it is a 'User Regression' operation). Let us start with a single parameter (mu_7) with second- and third-order self-interaction terms.

```
psuade> rscheck
Enter output number (1 - 2) = 1
Which response surface tool would you like?
0. MARS
1. Linear regression
2. Quadratic regression
3. Cubic regression
4. Quartic regression
5. Artificial Neural Network
6. User regression
Please enter your choice? 6
You may want to transform the output as follows:
0. no transformation.
1. log transformation on all the inputs.
2. log transformation on all the outputs.
3. log transformation on all inputs and outputs.
4. log transformation on selected inputs.
5. log transformation on selected inputs and outputs.
Please enter your choice? 0
RSFA:: output maximum norm          = 1.450000e-05
UserRegression INFO : using user_regression_file.
************************************************************
*************** User Regression Analysis *****************
* R-square gives a measure of the goodness of the model.   *
* R-square should be close to 1 if it is a good model.     *
*----------------------------------------------------------*
* UserRegression:: LS mean error =  8.3541e-08 (max= 1.4500e-05)
*----------------------------------------------------------*
*                     coefficient   std. error   t-value
* Constant         =  -7.7096e-04   4.1338e-04  -1.8650e+00
* Input  7         =   1.0974e-02   5.7759e-03   1.9000e+00
* Input  7   7     =  -5.1169e-02   2.6879e-02  -1.9037e+00
* Input  7   7   7 =   7.9577e-02   4.1660e-02   1.9101e+00
* UserRegression model R-square =   7.7960e-01
*==========================================================*
```

```
***** Standardized Regression Coefficients (SRC) ***********
* When R-square is acceptable (order assumption holds), the*
* absolute values of SRCs provide variable importance.     *
*----------------------------------------------------------*
* based on nSamples = 20
* Input  7         =   5.3341e+02
* Input  7   7     =  -2.2071e+01
*==========================================================*
PSUADE : refine check = 1.868e-08 <? 1.000e+00
```

Therefore, the conclusion from this exercise is that a third-order polynomial of parameter mu_7 can explain about 78% of the simulation data. Let us consider a regression model which consists of linear terms of all the parameters (i.e. Linear regression), quadratic and cubic polynomials for mu_7.

```
psuade> rscheck
Enter output number (1 - 2) = 1
Which response surface tool would you like ?
0. MARS
1. Linear regression
2. Quadratic regression
3. Cubic regression
4. Quartic regression
5. Artificial Neural Network
6. User regression
Please enter your choice ? 6
You may want to transform the output as follows:
0. no transformation.
1. log transformation on all the inputs.
2. log transformation on all the outputs.
3. log transformation on all inputs and outputs.
4. log transformation on selected inputs.
5. log transformation on selected inputs and outputs.
Please enter your choice ? 0
RSFA:: output maximum norm          = 1.450000e-05
UserRegression INFO : using user_regression_file.
************************************************************
**************** User Regression Analysis *****************
* R-square gives a measure of the goodness of the model.   *
* R-square should be close to 1 if it is a good model.     *
*----------------------------------------------------------*
* UserRegression:: LS mean error =  3.0645e-08 (max= 1.4500e-05)
*----------------------------------------------------------*
*                      coefficient   std. error   t-value
```

```
* Constant          =  -9.2130e-05   2.2183e-04  -4.1532e-01
* Input  7          =   1.4099e-03   3.1013e-03   4.5461e-01
* Input  7   7      =  -6.3701e-03   1.4429e-02  -4.4149e-01
* Input  7   7   7 =   9.7114e-03   2.2358e-02   4.3436e-01
* Input  1          =  -9.8357e-05   2.6853e-05  -3.6628e+00
* Input  2          =  -2.6728e-05   5.1262e-06  -5.2139e+00
* Input  3          =   1.2744e-03   2.5568e-04   4.9843e+00
* Input  4          =  -2.1207e-03   2.8164e-03  -7.5299e-01
* Input  5          =   5.9220e-06   1.4101e-06   4.1996e+00
* Input  6          =   2.1970e-06   1.0330e-06   2.1269e+00
* UserRegression model R-square =   9.7034e-01
*============================================================*
***** Standardized Regression Coefficients (SRC) ***********
* When R-square is acceptable (order assumption holds), the*
* absolute values of SRCs provide variable importance.     *
*------------------------------------------------------------*
* based on nSamples = 20
* Input  7          =   6.8531e+01
* Input  7   7      =  -2.7477e+00
* Input  1          =  -2.2298e-01
* Input  2          =  -3.3864e-01
* Input  3          =   3.4168e-01
* Input  5          =   2.8785e-01
* Input  6          =   1.4239e-01
*============================================================*
PSUADE : refine check = 6.853e-09 <? 1.000e+00
```

As comparing with result from regression model which only consists of linear term, the current model is better since the R-square value increases from 96.8% to 97.0%. The result from sensitivity analyses are summarized in the following Table:

Table 3 Summary of Sensitivity Analyses using Different Methods

Methods	Highly Sensitive	Marginal Sensitive	Not Sensitive
MARS	#2 and #7		
SRC - Main Effect	#7	#1, #2, #3, #5, #6	#4
SRC - User Regression	#7, $(\#7)^2$, see Remark	#1, #2, #3, #5, #6	#4

Remark: $(\#7)^2$ stands for second-order self-interaction term of parameter #7.

Example 3 –Uncertainty Quantification

For uncertainty quantification, one needs to create an animator file which points to the data file, e.g. `psuade010705.dat`. The basic idea is to use the data file as the surrogate model (response surface or metamodel) for uncertainty quantification. Appendix F lists the content of PSUADE interface program for uncertainty quantification.

As one can see this animator is basically an initial setup file for PSUADE (like example1 in Appendix B), except a few changes

- the number of sampling (num_samples) in METHOD being extended to be 500
- the name of surrogate model (driver) in APPLICATION section being specified,
- response type (analyzer rstype) in ANALYSIS section being specified, e.g. MARS, and
- the probability density function (PDF) of each input parameter in INPUT section being specified.

Four examples are presented hereafter. Table 3 lists the different options being used in these examples. The first 4 examples are to compare the result using different sampling scheme (MARS, LINEAR, and user_regression). The linear regression used in example 3-3 is based on a model consisting of all the first-order polynomial terms. The user_regression model used in examples 3-4 and 3-5 are based on the same data set but with additional regression terms (the second- and third-order self-interaction terms of parameter #7). Examples 3-4 and 3-5 are used to show the difference with different distribution profiles.

Table 4: List of Options used in Uncertainty Quantification

Example	rstype	PDF	num_samples
3-1	MARS	U (Uniform)	500
3-2	MARS	U (Uniform)	2000
3-3	LINEAR	U (Uniform)	500
3-4	User_regression	U (Uniform)	500
3-5	User_regression	N (Normal)	500

After these preparations, one can sample the response of the surrogate model using PSUADE to create sampling point (see Example 1: for creating design matrix and sampling points). In other words, one simply keys in ***psuade <filename>***. PSUADE will create a file named ***psuadeData***.

Once again, the user needs to rename the psuadeData so that it will not get overwritten. After the sampling points having been collected using surrogate model, one can proceed to study the probabilistic structure of the response. For uncertainty quantification, user shall launch psuade and then give follow the instruction:

```
*** ************************************************** ***
*** Welcome to PSUADE main program (version 1.0)       ***
*** ************************************************** ***
PSUADE - A Problem Solving environment for
         Uncertainty Analysis and Design Exploration
psuade> load example4-4.dat
```

Let us load the data file which is generated based on the setting listed in Table 3.

```
load complete : nSamples = 500
psuade> uq
```

For uncertainty quantification, the user shall enter ***uq*** and then choose which output response to analyze.

```
Enter output number (1 - 2) = 1

*******************************************************
** Standard error of mean calculation               **
-------------------------------------------------------
* nSamples =        500                              **
* nGroups  =          1                              **
* nInputs  =          7                              **
* output   =          0                              **
-------------------------------------------------------
*       aggregate mean          =   1.45498694e-05
*       aggregate std dev       =   1.33582274e-08
*******************************************************
matlabuq.m is now available.
```

Notice that aggregate mean and std. dev. , stand for mean and standard deviation of the response surface using the surrogate model. A MATLAB program, which contains the histogram of the response surface, will be created at the end. One must rename the resultant MATLAB program so that it will not get overwritten and, in this example, proceed to study the second response:

```
psuade> uq
Enter output number (1 - 2) = 2
```

```
*******************************************************
** Standard error of mean calculation                **
-------------------------------------------------------
* nSamples =        500                              **
* nGroups  =          1                              **
* nInputs  =          7                              **
* output   =          1                              **
-------------------------------------------------------
*       aggregate mean          =   1.79751590e-05
*       aggregate std dev       =   2.64676937e-08
*******************************************************
matlabuq.m is now available.
suade> quit
```

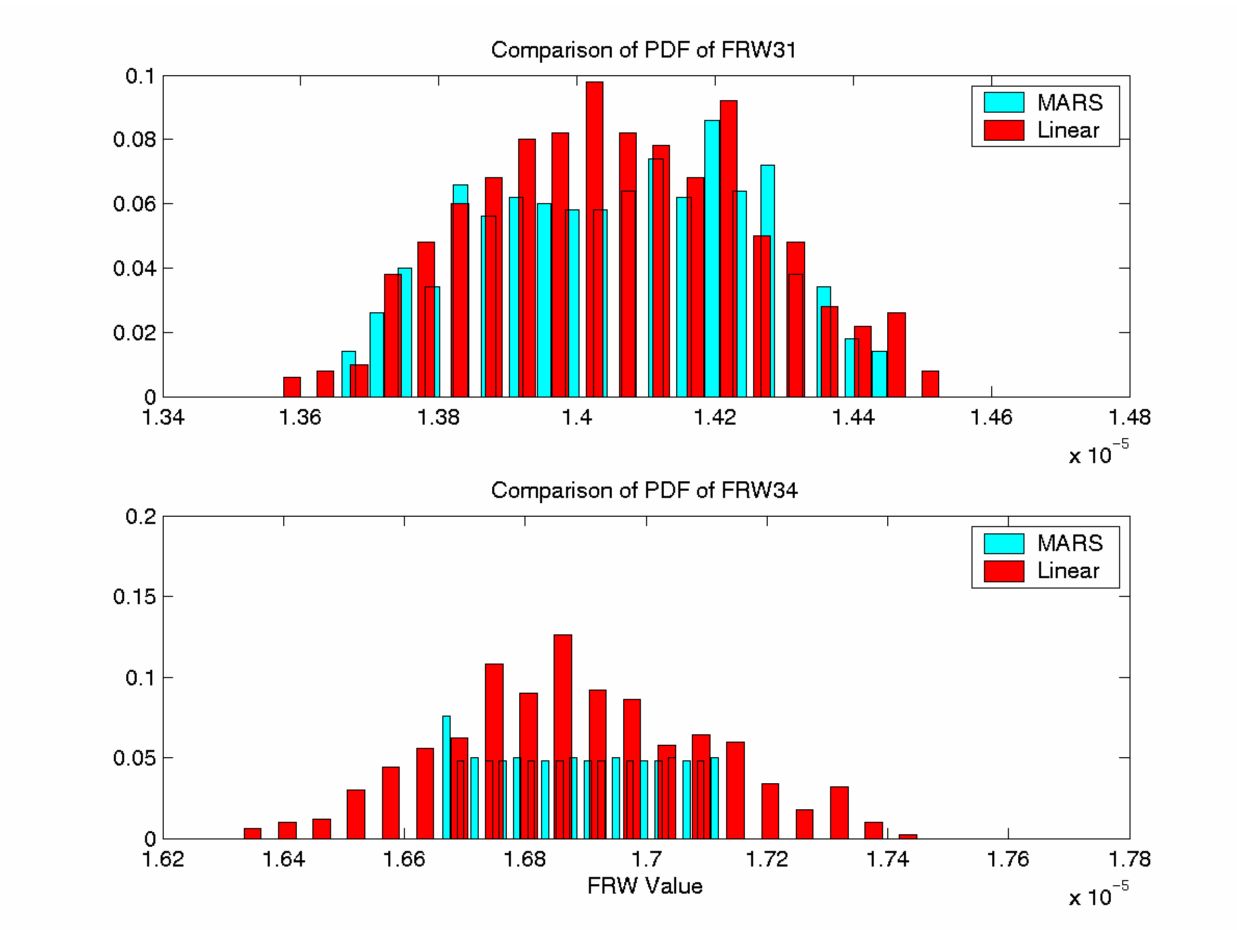

Figure 5: Comparison of PDF between Parametric (Linear) and Non-parametric (MARS) regression models.

Figure 5 shows the resultant probabilistic density function (PDF) generated using Linear and MARS regression models. Notice that MARS is a non-parametric regression model. There exists little difference between the PDF of FRW31 (frictional work at velocity =31 m/sec) generated based on linear parametric and MARS regression models. However, there exist some discrepancies between the PDF of FRW34 generated using linear parametric and MARS regression models. *This is an indication insufficient sample point in using MARS.*

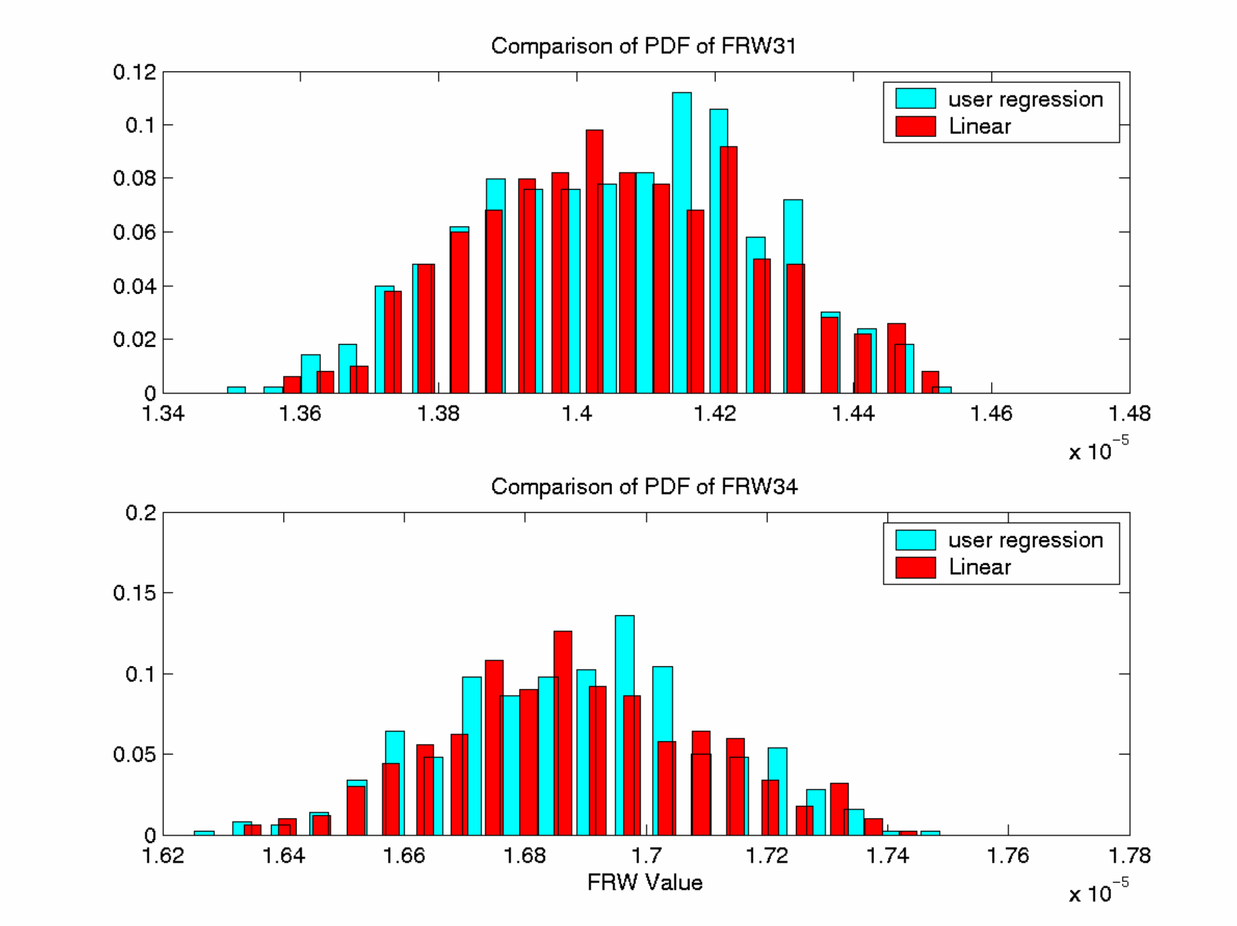

Figure 6: Comparison of PDF between Linear Parametric (Examples 4-3) and User Defined Regression Example 4-4) Models.

Figure 6 shows the comparison of PDF generated using linear parametric and user-defined regression models. This purpose of this comparison is merely to show the effect of the second- and third-order self-interaction effects of parameter #7. The high-order self-interaction terms cause the spread out of the PDF.

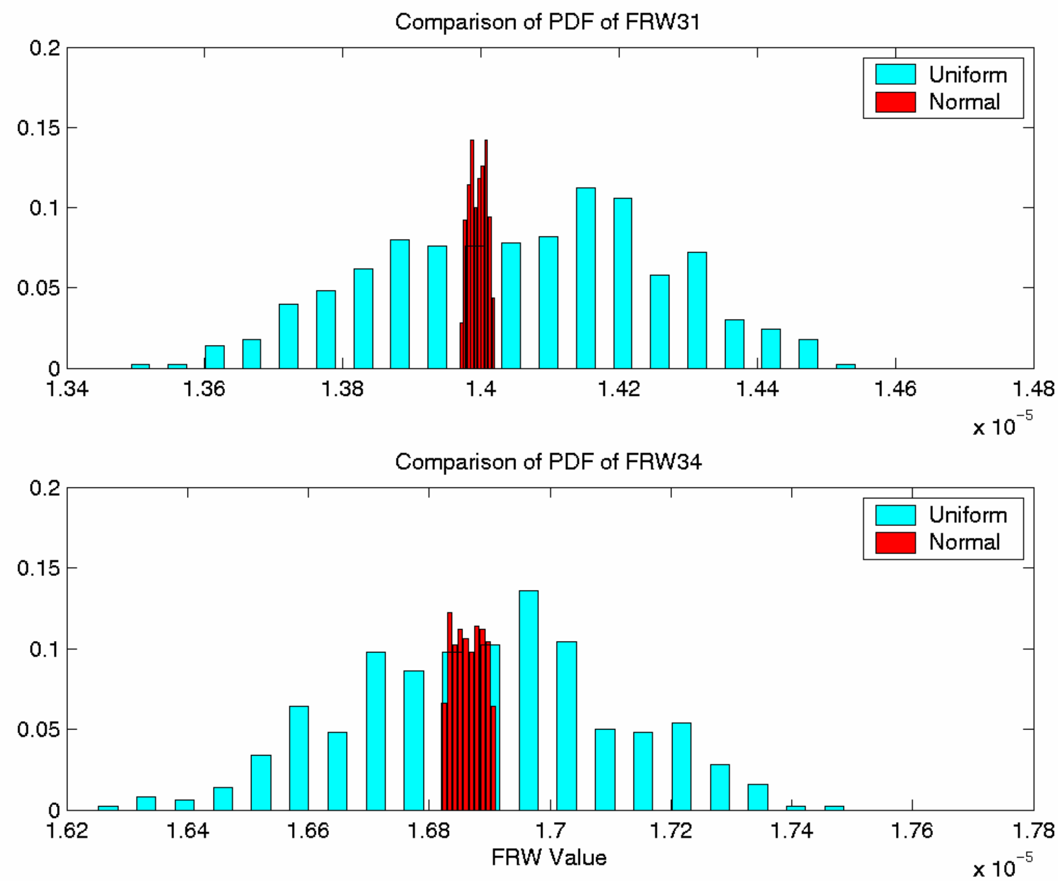

Figure 7: Comparison of PDF for Parameters with Uniform (Example 4-4) against Normal Distributions (Example 4-5).

Figure 7 compares the PDF generated using same user defined regression model with different probability distribution functions. From Figure 7, it is clear that uniform distribution will result in a broader PDF. A potential usage of this is for model correlation, model calibration, and model validation (with experimental data being available). Since the user defined regression model which consists of second- and third-order self-interaction terms, the resultant PDF will *not* be of Gaussian distribution.

6. Conclusion

Figure 8 shows a general flowchart for creation of a metamodel (response surface or surrogate model). The process consists of four basic steps: experimental design, model choice, model fitting, and sample approximation techniques. Depending upon the choice selected in each step, different metamodels can be formed to approximate the functional relationship between input and output factors of a simulation model. For example, one can use Latin-Hypercube as the sampling technique, Gaussian Process (realization of a stochastic process) as the model, and then apply Best Linear Unbiased Predictor to fit a Kriging model. Alternatively, the analyst can use the same sampling points, choose lower-order polynomial functions as the model, and then apply least-square regression to find a traditional response surface. Items highlighted with red color in Figure 8 are available in the version of PSUADE (1.0) used in writing this report.

Experimental Design	**Model Choice**	**Model Fitting**	**Sample Approximation Techniques**
(Fractional) Factorial	Polynomial (linear, quadratic,..)	Least Square Regression	Traditional Response Surface
Central Composite	Splines (Linear, Cubic)	Weighted Least Square Regression	
Box-Behnken	Realization of a Stochastic Process	Best Linear Unbiased Predictor	Kriging
D-Optimal	Radial Basis Functions	Entropy	
G-Optimal	Network of Neurons		Arbitrary Neural Networks
Orthogonal Array			
Latin Hypercube			
Random Selection			

Figure 8: General Flowchart in Creating Response Surface

Using the Steven Impact Test as the example, we applied some of the analysis capabilities to show different methods of sensitivity analysis and uncertainty quantification. It is possible to receive different sensitivity analysis result if one chooses different assessment methods. Table 3 summarized sensitivity analysis results using different methods. Since the sensitivity analysis results are consistent, we used the same data set for uncertainty quantification. A user must be alert if qualitative discrepancies in sensitivity analyses appear. This is an indication of either bad practice in design exploration or some errors in data analysis.

There are other software packages, such as NESSUS[2], DAKOTA and iSIGHT[3], which provide similar functionality as PSUADE. PSUADE distinguishes itself from these tools in the following aspects:

- PSUADE is the only package, based on the best knowledge of the author, which offers the Morris-One-At-a-Time (MOAT) [14] scheme for screening experiment. MOAT is by far the most efficient design exploration method for screening problems with high dimension, e.g., the number of parameters (control and input factors together) exceeds 20. The traditional two-level, full factorial design for screening experiment is unfeasible when the number of parameters is greater than 10. When the number of parameters equals to 13, the numbers of analyses required for MOAT and a two-level resolution IV factorial design are about the same. With the same number of analyses, MOAT can provide a better diagnosis for screening purpose; whereas the traditional (fractional) factorial design will be crippled with the 'aliasing' effect [9, 10]. There is one tradeoff in using MOAT since the method cannot assess interaction effects quantitatively. However, MOAT does provide qualitative evidence to the user so that interaction effect may be explored later on. for example, during construction of the response surface. There are several successful application examples of MOAT in B Division.
- Another feature that distinguishes PSUADE from other tools is its "adaptive sampling". Beyond the curse of high dimension, another issue that analysts often face is lack of knowledge prior to analysis. Most approaches to approximate a system's response make step-by-step improvements to the approximation model by adjusting the limits of the

[2] NESSUS is a registered trademark of Southwest Research Institute.
[3] iSIGHT is a registered trademark of Engineous Software, Inc.

design variables. One way to improve the efficiency of response surface construction is to apply adaptive sampling. PSUADE has implemented such capabilities so the user can start with a small number of computer experiments and then, if needed, refine the original design space to achieve a better resolution. In other words, PSUADE can help users optimize their analysis by reducing the number of analyses.

- Unlike iSIGHT and Dakota, whose initial development aimed for design optimization, PSUADE is developed to facilitate global/local sensitivity and uncertainty quantification. Moreover, PSUADE is developed to wrap around the simulation software already used in Livermore Lab.
- As compared with NESSUS, whose development aimed for reliability assessment, PSUADE offers a much better collection of design of experiments, sensitivity analysis, and uncertainty quantification capabilities. As far as reliability assessment is concerned, the future release of PSUADE will have traditional analytical methods for reliability assessment, such First-Order Reliability Method (FORM) and Second-Order Reliability Method (SORM).

Based upon the author's experience, the following are needed to facilitate the application of PSUADE for probabilistic analyses:

- A graphic user interface (GUI) for the analyst. The current version of PSUADE is very much command line driven. A GUI will facilitate the communication between user and PSUADE. At the present time, PSUADE does not have its own graphing package to display the analysis result. An extension of PSUADE to utilize existing graphics packages, such as VisIt, will also enhance analysts efficiency.
- Some enhancement on conventional statistics analysis and graphical display of analysis result, such as provided by MiniTab[4]. MiniTab is a Microsoft Excel-based commercial software for traditional statistics.
- An extensive library of probability density (distribution) functions for continuous (discrete) random variables in assessing uncertainty. On this aspect, Crystal Ball[5][17] offers a board choice of probability functions to users. Crystal Ball, is a spread sheet add-on that performs uncertainty, response surface and Monte Carlo forecasting analyses.

[4] Minitab is a registered trademark of Minitab, Inc.
[5] Crystal Ball is a registered trademark of Decisioneering, Inc.

References:

1. Magness, F. H.,"*An Assessment of the Frictional Work Method and Its Use as an HE Ignition Predictor in Small-Scale Impact Test Configurations*", CODTU-2004-1850.
2. Garcia, M., "*Frictional Work and Ignition and Growth modeling of PBX 9404 in the Steven and Susan Impact Tests (U)* "", CODTU-2004-1806.
3. Vandersall, K.S., Murty, S. S., Chidester, S. K., Forbes, J. W., Garcia F., Greenwood, D. W., and Tarver, C. M., "*Investigation of Steven Impact Test Using a Transportation Hook Projectile with Gauged Experiments and 3D Modeling*", Preprint UCRL-JC-152303
4. Vandersall, K.S., Murty, S. S., Chidester, S. K., Forbes, J. W., Garcia F., Greenwood, D. W., and Tarver, C. M., " *Experimental and Modeling Studies of Crush, Puncture, and Perforation Scenarios in the Steven Impact Test*", 12th International Detonation Symposium proceedings, Meeting held August 11-16, 2002
5. DePiero, A., "*Methodology for Predicting HE Ignition due to Low-Pressure, High-Deformation Impacts*", preprint, CODTU-00-0149.
6. Logan, R., Nitta, C., Childester, S., "*Risk Reduction as the Product of Model Assessed Reliability, Confidence, and Consequence*", UCRL-AR-200703.
7. Hoffman, D. M. and Chandler, J. B., "*Aspects of the Tribology of the Plastic Bonded Explosive (PBX) 9404*", Preprint, UCRL-JRNL-205202
8. Myers, R.H., *Classical and Modern Regression with Application,* Duxbury Press; 2 edition, 2000, ISBN: 0534380166
9. Montgomery, D.C., *Design and Analysis of Experiments*, John Wiley & Sons; 6 edition 2004, ISBN: 047148735X
10. Meyers, R. H. and Montgomery, D. C., *Response Surface Methodology : Process and Product Optimization Using Designed Experiments*, Wiley-Interscience; 2 edition, 2002, ISBN: 0471412554
11. Saltelli A. Tarantola S., Campolongo, F. and Ratto, M., *Sensitivity Analysis in Practice. A Guide to Assessing Scientific Models*, John Wiley & Sons publishers, 2004.
12. Saltelli, A., Chan, K., and Scott, M., *Sensitivity Analysis*, John Wiley & Sons publishers, 2000.
13. Friedman, J. H.,"*Multivariate Adaptive Regression Splines (with discussion)*", Annals of Statistics, 1991, Volume 19, page 1-141.
14. Morris, M. D., "Factorial Sampling Plans for Preliminary Computational Experiments", *Technometrics*, 1991, Volume 33, page 61-174.
15. Tong, Charles, 2005, *PSUADE User Manual.*
16. Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K., "*Metamodels for Computer-based Engineering Desing: Survey and Recommendations*", Engineering with Computers, 2001, Volume 17, page 129-150.
17. Decisioneering, Inc., *Crystal Ball User Manual*

Appendix A: Python script

```
#!/usr/local/bin/python
# sys.argv[1] - input file
# sys.argv[2] - output file

# import python standard libraries
# os - for operation system interface
import os
import sys
import string
import shutil
# The glob module provides a function for making file
# lists from directory wildcard searches:
import glob


##########################################################################
##########################################################################
# Function to get input data from PSUADE input file
#-------------------------------------------------------------------------
def getInputData(inFileName, nInputs):
   if nInputs <= 0:
      inputData = []
      return inputData
   inFile = open(inFileName, "r")
   inputData = range(nInputs)
   lineIn = inFile.readline()
   count = 0
   while 1:
      lineIn = inFile.readline()
      ncols = string.split(lineIn)
      inputData[count] = eval(ncols[0])
      count = count + 1
      if count >= nInputs:
         break
   return inputData

##########################################################################
# Function to generate input file for the application
#-------------------------------------------------------------------------
def genApplicationInputFile(inputData,appTmpltFile,appInputFile,nInputs,
                            inputNames):
   infile = open(appTmpltFile, "r")
   outfile = open(appInputFile, "w")
   while 1:
      lineIn  = infile.readline()
      if lineIn == "":
         break
      lineLen = len(lineIn)
      newLine = lineIn
      if nInputs > 0:
         for fInd in range(nInputs):
            strLen = len(inputNames[fInd])
            sInd = string.find(newLine, inputNames[fInd])
```

```
            if sInd >= 0:
               strdata = str(inputData[fInd])
               next = sInd + strLen
               lineTemp = newLine[0:sInd] + strdata + newLine[next:lineLen]
               newLine = lineTemp
      outfile.write(newLine)
   infile.close()
   outfile.close()
   return

###########################################################################
# Function to generate the batch file
#--------------------------------------------------------------------------
def genBatchFile(batchTmpltFileName, batchFileName):
   infile = open(batchTmpltFileName, "r")
   outfile = open(batchFileName, "w")
   searchString = "PSUADE_COUNTER"
   strLen = len(searchString)
   while 1:
      lineIn  = infile.readline()
      if lineIn == "":
         break
      lineLen = len(lineIn)
      newLine = lineIn
      sInd    = string.find(newLine, searchString)
      if sInd >= 0:
         next = sInd + strLen
         strdata = str(fileTag)
         lineTemp = newLine[0:sInd] + strdata + newLine[next:lineLen]
         newLine = lineTemp
      outfile.write(newLine)
   infile.close()
   outfile.close()
   return

###########################################################################
# Function to run batch file
#--------------------------------------------------------------------------
def runApplication(batchFileName):
   sysCommand = "/usr/local/bin/psub " + batchFileName
   os.system(sysCommand)
   statCommand = "/usr/local/bin/pstat | /bin/grep " + batchFileName
   status = os.system(statCommand)
   while status == 0:
      os.system("sleep 60")
      status = os.system(statCommand)
   return

###########################################################################
# Function to get output file from the application
#--------------------------------------------------------------------------
def getApplicationOutputData():
   searchString = "OUTPUT"
   infile = open("ssdns.info", "r")
   outData = range(1)
   while 1:
      lineIn = infile.readline()
```

```
      if lineIn == "":
         break
      sInd = string.find(lineIn, searchString)
      if sInd >= 0:
         ncols = string.split(lineIn)
         outData[0] = eval(ncols[1])
   infile.close()
   return outData

##########################################################################
# Function to generate output file from the application
#-------------------------------------------------------------------------
def genOutputFile(outFileName, outData):
   outfile = open(outFileName, "w")
   outfile.write("%e \n" % outData[0])
   outfile.write("%e \n" % outData[1])
   outfile.close()
   return


##########################################################################
## Main program
##########################################################################

# -----------------------------------------------------------------------
# fetch PSUADE input and output file names
# -----------------------------------------------------------------------
inFileName = sys.argv[1]
outFileName = sys.argv[2]

# -----------------------------------------------------------------------
# application specific settings
# -----------------------------------------------------------------------
testFlag = 1
appInputTmpltFile = "input.Tmplt"
batchTmpltFile = "batch.Tmplt"
nInputs = 7
inputNames = ["kkkkkk", "gggggg", "sigma_y", "eheheh", "mu_3", "mu_6",
"mu_7"]
copyList = [appInputTmpltFile, batchTmpltFile, inFileName]

print copyList
# -----------------------------------------------------------------------
# if output file already exists, do not perform test
# -----------------------------------------------------------------------
if os.path.isfile(outFileName) != 0:
   testFlag = 0
   exit

# -----------------------------------------------------------------------
# extract the sample number -> fileTag
# -----------------------------------------------------------------------
tagString = os.path.splitext(inFileName)[1]
# split the pathname path into a pair (root, ext) such that
# root + ext == path, and ext is empty or begins with a period and
# contains at most one period
# here is to extract the extension
```

```
stringLen = len(tagString)
fileTag   = eval(tagString[1:stringLen])
# --------------------------------------------------------------------------
# If working directory already exists, do not perform test. Otherwise,
# create the working directory
# --------------------------------------------------------------------------
dirname = "./workdir." + str(fileTag)
if os.path.isdir(dirname):
   testFlag = 0
else:
   os.mkdir(dirname)

# --------------------------------------------------------------------------
# copy the needed files to the working directory and enter into it
# --------------------------------------------------------------------------
if testFlag == 1:
   for file in copyList:
      shutil.copy(file, dirname)
# copyfile( src, dst)
# copy the contents of the file named src to a file named dst
   os.chdir(dirname)
# change to working directory - dirname
# --------------------------------------------------------------------------
# generate input file, run test and gather results
# --------------------------------------------------------------------------
if testFlag == 1:
   inputData = getInputData(inFileName, nInputs)
   appInputFile = os.path.splitext(appInputTmpltFile)[0] + '.' + str(fileTag)
   genApplicationInputFile(inputData,appInputTmpltFile,appInputFile,nInputs,
                           inputNames)
   batchFile = os.path.splitext(batchTmpltFile)[0] + '.' + str(fileTag)
   genBatchFile(batchTmpltFile, batchFile)
   runApplicationSrun(fileTag)
   outData = getApplicationOutputData()
   genOutputFile(outFileName, outData)
   shutil.copy(outFileName, "..")
   for file in glob.glob('*'):
      os.remove(file)
   os.chdir("..")
   os.rmdir(dirname)
```

Appendix B: Content of example1

```
PSUADE
INPUT
   dimension = 7
   variable   1 k =   6.29640000e-03   7.69560000e-03
   variable   2 g =   3.51900000e-02   4.30100000e-02
   variable   3 sigma_y =   7.44660000e-04   9.10140000e-04
   variable   4 eh =   5.79600000e-05   7.09400000e-05
   variable   5 mu_3 =   2.00000000e-01   2.30000000e-01
   variable   6 mu_6 =   2.30000000e-01   2.70000000e-01
   variable   7 mu_7 =   2.00000000e-01   2.30000000e-01
END
OUTPUT
   dimension = 2
   variable   1 frw31
   variable   2 frw34
END
METHOD
   sampling = LH
   num_samples = 20
END
APPLICATION
   driver = NONE
END
ANALYSIS
   diagnostics
END
END
```

Appendix C: Content of psuadeData

```
PSUADE_IO (Note : inputs not true inputs if pdf ~=U)
7 2 20 ← numbers of inputs, output responses, and sampling points
1 0            ← the first sampling point and the response readiness flag (not ready yet, since it is 0)
  7.1064631578947367e-03
  3.8071052631578950e-02
  9.0143052631578950e-04
  6.4791578947368423e-05    Input parameters
  2.0631578947368423e-01                                 one sampling point
  2.5105263157894736e-01
  2.0315789473684212e-01
  9.9999999999999997e+34    Output responses
  9.9999999999999997e+34
2 0
  6.5173263157894731e-03
  3.6836315789473682e-02
  7.6207894736842103e-04
  6.3425263157894737e-05
  2.2684210526315790e-01
  2.5315789473684214e-01
  2.3000000000000001e-01
  9.9999999999999997e+34
  9.9999999999999997e+34

.
. (skid on purpose to save space)
.
19 0
  6.9591789473684215e-03
  4.2598421052631577e-02
  8.3175473684210521e-04
  6.0692631578947365e-05
  2.0000000000000001e-01
  2.6789473684210530e-01
  2.2684210526315790e-01
  9.9999999999999997e+34
  9.9999999999999997e+34
20 0
  7.6219578947368423e-03
  4.1775263157894739e-02
  7.5336947368421050e-04
  6.5474736842105266e-05
  2.2368421052631579e-01
  2.5526315789473686e-01
  2.1578947368421053e-01
  9.9999999999999997e+34
  9.9999999999999997e+34
PSUADE_IO
PSUADE
INPUT
   dimension = 7
   variable 1 k  =   6.2963999999999997e-03   7.6956000000000004e-03
   variable 2 g  =   3.5189999999999999e-02   4.3010000000000000e-02
   variable 3 sigma_y  =   7.4465999999999996e-04   9.1014000000000004e-04
```

```
   variable 4 eh   =   5.7960000000000001e-05   7.0939999999999995e-05
   variable 5 mu_3  =   2.0000000000000001e-01   2.3000000000000001e-01
   variable 6 mu_6  =   2.3000000000000001e-01   2.7000000000000002e-01
   variable 7 mu_7  =   2.0000000000000001e-01   2.3000000000000001e-01
   PDF 1 U
   PDF 2 U
   PDF 3 U
   PDF 4 U
   PDF 5 U
   PDF 6 U
   PDF 7 U
```

probability density function of each input parameter is specified her

```
END
OUTPUT
   dimension = 2
   variable 1 frw31
   variable 2 frw34
```

2 output responses (frw31 and frw34)

```
END
METHOD
#  sampling = MC
#  sampling = FACTORIAL
   sampling = LH
#  sampling = OA
#  sampling = OALH
#  sampling = MCOAT
#  sampling = LHOAT
#  sampling = MOAT
#  sampling = SALTELLI
#  sampling = LPTAU
#  sampling = METIS
#  sampling = FAST
#  sampling = BBD
#  sampling = PBD
   num_samples = 20
   num_replications = 1
   num_refinements = 0
   reference_num_refinements = 0
#  randomize
```

Method section lists all the different methods available for user to choose from

```
#  randomize_more
END
APPLICATION
   driver = NONE
   opt_driver = NONE
   aux_opt_driver = NONE
   input_template = steven.k
   output_template = steven.out
   max_parallel_jobs = 4
   min_job_wait_time = 240
   max_job_wait_time = 6000
#  nondeterministic
#  launch_only
#  limited_launch_only
#  launch_interval = 15
#  save_frequency = 1000000
END
ANALYSIS
#  analyzer method = SEM
```

```
#  analyzer method = OneParamStudy
#  analyzer method = TwoParamStudy
#  analyzer method = ANOVA
#  analyzer method = GLSA
#  analyzer method = Regression
#  analyzer method = RSFA
#  analyzer method = MOAT
#  analyzer method = SALTELLI
#  analyzer method = Correlation
#  analyzer method = Integration
#  analyzer method = FAST
   analyzer outputID  = 1
   analyzer rstype = MARS
#  analyzer rstype = linear
#  analyzer rstype = quadratic
#  analyzer rstype = cubic
#  analyzer rstype = quartic
#  analyzer rstype = ANN
#  analyzer rstype = user_regression
#  graphics
#  sampleGraphics
   analyzer threshold = 1.000000e+00
#  optimization method = crude
#  optimization method = txmath
#  optimization method = appspack
#  optimization method = minpack
#  optimization method = cobyla
#  optimization method = sm
#  optimization method = mm
#  optimization num_local_minima = 0
#  optimization user_response_surface
#  optimization print_level = 0
#  optimization num_fmin = 0
#  optimization outputID = 0
#  optimization fmin = not defined
#  optimization cutoff = not defined
   diagnostics
#  fileWrite matlab
END
END
```

Appendix D: Content of pusade010705.dat

```
PSUADE_IO (Note : inputs not true inputs if pdf ~=U)
7 2 20
   1  1
0.766060E-02
0.361675E-01
0.889450E-03
0.621460E-04
0.220300E+00
0.257000E+00
0.209800E+00
0.141000E-04
0.169000E-04
   2  1
0.647130E-02
0.389045E-01
0.748800E-03
0.705180E-04
0.226300E+00
0.243000E+00
0.205300E+00
0.139000E-04
0.167000E-04
      .
      .
      .  (skid on purpose to save space)
      .
  19  1
0.661120E-02
0.377315E-01
0.897730E-03
0.666540E-04
0.202300E+00
0.259000E+00
0.212800E+00
0.141000E-04
0.170000E-04
  20  1
0.738080E-02
0.412505E-01
0.839810E-03
0.647220E-04
0.229300E+00
0.231000E+00
0.224800E+00
0.142000E-04
0.170000E-04
PSUADE_IO
PSUADE
INPUT
   dimension = 7
   variable 1 k  =   6.2963999999999997e-03   7.6956000000000004e-03
   variable 2 g  =   3.5189999999999999e-02   4.3010000000000000e-02
   variable 3 sigma_y  =   7.4465999999999996e-04   9.1014000000000004e-04
   variable 4 eh  =   5.7960000000000001e-05   7.0939999999999995e-05
```

notice the change of the second integer from 0 to 1 to indicate that output response is ready

output responses (frw31 and frw34)

```
   variable 5 mu_3  =   2.0000000000000001e-01   2.3000000000000001e-01
   variable 6 mu_6  =   2.3000000000000001e-01   2.7000000000000002e-01
   variable 7 mu_7  =   2.0000000000000001e-01   2.3000000000000001e-01
   PDF 1 U
   PDF 2 U
   PDF 3 U
   PDF 4 U
   PDF 5 U
   PDF 6 U
   PDF 7 U
END
OUTPUT
   dimension = 2
   variable 1 frw31
   variable 2 frw34
END
METHOD
#  sampling = MC
#  sampling = FACTORIAL
   sampling = LH
#  sampling = OA
#  sampling = OALH
#  sampling = MCOAT
#  sampling = LHOAT
#  sampling = MOAT
#  sampling = SALTELLI
#  sampling = LPTAU
#  sampling = METIS
#  sampling = FAST
#  sampling = BBD
#  sampling = PBD
   num_samples = 20
   num_replications = 1
   num_refinements = 0
   reference_num_refinements = 0
#  randomize
#  randomize_more
END
APPLICATION
   driver = NONE
   opt_driver = NONE
   aux_opt_driver = NONE
   input_template = steven.k
   output_template = steven.out
   max_parallel_jobs = 4
   min_job_wait_time = 240
   max_job_wait_time = 6000
#  nondeterministic
#  launch_only
#  limited_launch_only
#  launch_interval = 15
#  save_frequency = 1000000
END
ANALYSIS
#  analyzer method = SEM
#  analyzer method = OneParamStudy
#  analyzer method = TwoParamStudy
```

```
#  analyzer method = ANOVA
#  analyzer method = GLSA
#  analyzer method = Regression
#  analyzer method = RSFA
#  analyzer method = MOAT
#  analyzer method = SALTELLI
#  analyzer method = Correlation
#  analyzer method = Integration
#  analyzer method = FAST
   analyzer outputID  = 1
   analyzer rstype = MARS
#  analyzer rstype = linear
#  analyzer rstype = quadratic
#  analyzer rstype = cubic
#  analyzer rstype = quartic
#  analyzer rstype = ANN
#  analyzer rstype = user_regression
#  graphics
#  sampleGraphics
   analyzer threshold = 1.000000e+00
#  optimization method = crude
#  optimization method = txmath
#  optimization method = appspack
#  optimization method = minpack
#  optimization method = cobyla
#  optimization method = sm
#  optimization method = mm
#  optimization num_local_minima = 0
#  optimization user_response_surface
#  optimization print_level = 0
#  optimization num_fmin = 0
#  optimization outputID = 0
#  optimization fmin = not defined
#  optimization cutoff = not defined
   diagnostics
#  fileWrite matlab
END
END
```

Appendix E: user_regression_file

```
9                how many terms are defined in the user regression file
1 1 1
2 1 2
3 1 3
4 1 4            linear order term
5 1 5
6 1 6
7 1 7
8 2 7 7          second-order self-interaction term
9 3 7 7 7        third order self-interaction term
```

Appendix F: Content of Example3-1

```
PSUADE
INPUT
   dimension = 7
   variable   1 k =   6.29640000e-03   7.69560000e-03
   variable   2 g =   3.51900000e-02   4.30100000e-02
   variable   3 sigma_y =   7.44660000e-04   9.10140000e-04
   variable   4 eh =   5.79600000e-05   7.09400000e-05
   variable   5 mu_3 =   2.00000000e-01   2.30000000e-01
   variable   6 mu_6 =   2.30000000e-01   2.70000000e-01
   variable   7 mu_7 =   2.00000000e-01   2.30000000e-01
   PDF 1 U
   PDF 2 U
   PDF 3 U
   PDF 4 U
   PDF 5 U
   PDF 6 U
   PDF 7 U
END
OUTPUT
   dimension = 2
   variable   1 frw31
   variable   2 frw34
END
METHOD
   sampling = LH
   num_samples = 500
END
APPLICATION
   driver = psuade010705.dat
   input_template = NONE
   output_template = NONE
   max_parallel_jobs = 4
   min_job_wait_time = 240
   max_job_wait_time = 6000
END
ANALYSIS
   analyzer outputID = 1
   analyzer rstype = MARS
END
END
```